LISTING OF CLAIMS:

1. (previously presented) A method in a data processing system for monitoring a plurality of related threads, the method comprising the data processing system implemented steps of:

polling the plurality of related threads for status information;

responsive to receiving the status information, determining whether a thread within a plurality of related threads is inactive; and

responsive to a determination that a thread within the plurality of related threads is inactive, initiating cleanup processes for the thread based on the status information.

B1

2. (original) The method of claim 1 further comprising:

responsive to receiving the status information, storing the status information.

3. (original) The method of claim 1, wherein the polling, determining, and initiating steps are performed by a single thread.

4. (previously presented) The method of claim 1, wherein the single thread is part of a class.

5. (original) The method of claim 1, wherein the initiating step comprises:

identifying active threads within the plurality of related threads;

identifying inactive threads within the plurality of related threads; and

terminating inactive threads.

6. (canceled)

7. (original) The method of claim 1, wherein the plurality of related threads is a plurality of printer threads.

8. (original) The method of claim 1, wherein the plurality of related threads is a plurality of video threads.

PAGE 4/14 * RCVD AT 12/24/2003 11:06:01 AM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:9723672002 * DURATION (mm-ss):04-18

9. (original) The method of claim 1, wherein the method is implemented in a virtual machine.

10. (original) The method of claim 9, wherein the virtual machine is a Java virtual machine.

11. (previously presented) A method in a data processing system for monitoring a plurality of related threads, the method comprising the data processing system implemented steps of:

polling the plurality of related threads for status information;

responsive to receiving the status information, determining whether a thread within a plurality of related threads is inactive; and

responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, initiating cleanup processes based on the status information.

12. (previously presented) The method of claim 11, wherein the event is an occurrence of a period of time.

13. (original) The method of claim 11, wherein the event is an error.

14. (previously presented) A data processing system for monitoring a plurality of related threads, the data processing system comprising:

polling means for polling the plurality of related threads for status information;

determining means, responsive to receiving the status information, for determining whether a thread within a plurality of related threads is inactive; and

initiating means, responsive to a determination that a thread within the plurality of related threads is inactive, for initiating cleanup processes for the thread based on the status information.

15. (original) The data processing system of claim 14 further comprising:

storing means, responsive to receiving the status information, for storing the status information.

PAGE 5/14 * RCVD AT 12/24/2003 11:06:01 AM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:9723672002 * DURATION (mm-ss):04-18

16. (original) The data processing system of claim 14, wherein the polling, determining, and initiating means are preformed by a single thread.

17. (previously presented) The data processing system of claim 14, wherein the single thread is part of a class.

18. (original) The data processing system of claim 14, wherein the initiating means comprises:

first identifying means for identifying active threads within the plurality of related threads;

second identifying means for identifying inactive threads within the plurality of related threads; and

terminating means for terminating inactive threads.

19. (canceled)

20. (original) The data processing system of claim 14, wherein the plurality of related threads is a plurality of printer threads.

21. (original) The data processing system of claim 14, wherein the plurality of related threads is a plurality of video threads.

22. (original) The data processing system of claim 14, wherein the data processing system is implemented in a virtual machine.

23. (original) The data processing system of claim 22, wherein the virtual machine is a Java virtual machine.

24. (previously presented) A data processing system for monitoring a plurality of related threads, the data processing system comprising:

polling means for polling the plurality of related threads for status information;

PAGE 6/14 * RCVD AT 12/24/2003 11:06:01 AM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:9723672002 * DURATION (mm-ss):04-18

determining means, responsive to receiving the status information, for determining whether a thread within a plurality of related threads is inactive; and

initiating means, responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, for initiating cleanup processes based on the status information.

25. (previously presented) The data processing system of claim 24, wherein the event is an occurrence of a period of time.

B1

26. (original) The data processing system of claim 24, wherein the event is an error.

27. (previously presented) A computer program product in a computer readable medium for monitoring a plurality of related threads, the computer program product comprising:

first instructions for polling the plurality of related threads for status information;

second instructions for responsive to receiving the status information, determining whether a thread within a plurality of related threads is inactive; and

third instructions for responsive to a determination that a thread within the plurality of related threads is inactive, initiating cleanup processes for the thread based on the status information.

28. (previously presented) A computer program product in a computer readable medium for monitoring a plurality of related threads, the computer program product comprising:

first instructions for polling the plurality of related threads for status information;

second instructions, responsive to receiving the status information, for determining whether a thread within a plurality of related threads is inactive; and

third instructions, responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, for initiating cleanup processes based on the status information.

PAGE 7/14 * RCVD AT 12/24/2003 11:06:01 AM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:9723672002 * DURATION (mm-ss):04-18